Реализация абстракции в ООП. Абстрактные классы и интерфейсы в Java

1) Абстракция в программировании – подход, который даёт возможность при проектировании и реализации приложений оперировать не конкретными объектами (терминами, вещами, компонентами и т.д.) либо отдельными их характеристиками (свойствами) или действиями (поведением), а их общими (собирательными) понятиями, которых в реальном мире не существуют либо существуют частично.

2) Абстрактное поведение – поведение, о котором известно, что оно должно делать, но не известно, как оно реализовано. Каждый, кто хочет удовлетворять спецификации (интерфейсу) данного абстрактного поведения, должен самостоятельно его реализовать.

3) Основная концепция абстрактного класса в Java – реализовать интерфейс – абстрактную независимую прослойку – между конкретными компонентами-исполнителями определённого действия и компонентами-потребителями результатов данного действия.

4) Абстрактный класс содержит следующие программные элементы: статические и динамические поля, статические и динамические блоки инициализации, конструкторы, статические, динамические и абстрактные методы, описание других типов данных (классы, интерфейсы и перечисления), описание статических классов.

5) Абстрактный класс отличие от обычного класса лишь тем, что в абстрактном классе можно дополнительно описать абстрактное поведение в виде абстрактных методов.

6) В Java поддерживается множественное расширение между интерфейсами, поддерживается множественная реализация интерфейсов классами, а вот множественное наследование классов в Java не реализовано.

7) Основная концепция отделения интерфейса от реализации – независимость компонентов друг от друга, гибкость и масштабируемость архитектуры приложения.

8) …

9) …

10) …

11) …

12) …

13) …

14) …

Перечисляемый тип в Java (since JDK 5.0)

15) …

16) …

17) …

18) …

19) …

20) …

21) …

22) …

23) …

24) …

SOLID и GRASP принципы

25) …

26) …
27) …
28) …
29) …
30) …
31) …
32) …
33) …
34) …
35) …
36) …

Приёмы объектно-ориентированного проектирования. Шаблоны проектирования

37) …
38) …
39) …
40) …
41) …
42) …
43) …

Параметризация (обобщение) типов в Java – Java Generic (since JDK 5.0)

44) …
45) …
46) …
47) …
48) …
49) …
50) …
51) …
52) …
53) …
54) …
55) …

Библиотека контейнеров Java Collections Framework (since JDK 5.0)

56) …
57) …
58) …
59) …
60) …
61) …
62) …

63) …
64) …
65) …
66) …
67) …
68) …
69) …
70) …
71) …
72) …
73) …
74) …
75) …
76) …
77) …
78) …
79) …
80) …
81) …
82) …
83) …
84) …
85) …
86) …

Основы потоков ввода-вывода в Java

87) …
88) …
89) …
90) …
91) …
92) …
93) …
94) …
95) …
96) …
97) …
98) …
99) …
100) …
101) …

102) …
103) …
104) …
105) …
106) …
107) …
108) …
109) …

Основы параллельного (многопоточного) программирования в Java. Потоки выполнения

110) …
111) …
112) …
113) …
114) …
115) …
116) …
117) …
118) …
119) …
120) …

Основы синхронизации потоков в Java. Использование библиотеки java.util.concurrency

121) …
122) …
123) …
124) …
125) …
126) …
127) …
128) …
129) …
130) …